BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

EXPRESS MAIL NO. EV301024810US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Yolanta BERESNEVICHIENE, et al.
Serial No.: Not yet assigned
Filed: Concurrently herewith
For: "IMPROVEMENTS IN AND RELATING TO COMPUTER OPERATING SYSTEM DATA MANAGEMENT"

Our Ref: B-5366 621683-7

Date: January 26, 2004

CLAIM TO PRIORITY UNDER 35 U.S.C. 119

MAIL STOP PATENT APPLICATION
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

[X] Applicants hereby make a right of priority claim under 35 U.S.C. 119 for the benefit of the filing date(s) of the following corresponding foreign application(s):

COUNTRY	FILING DATE	SERIAL NUMBER
GB	27 January 2003	0301777.9

[ ] A certified copy of each of the above-noted patent applications was filed with the Parent Application No._____.

[X] To support applicants' claim, a certified copy of the above-identified foreign patent application is enclosed herewith.

[ ] The priority document will be forwarded to the Patent Office when required or prior to issuance.

Respectfully submitted,

Richard P. Berg
Attorney for Applicant
Reg. No. 28,145

LADAS & PARRY
5670 Wilshire Boulevard
Suite 2100
Los Angeles, CA 90036
Telephone: (323) 934-2300
Telefax: (323) 934-0202

EV301024810US

PATENTS • DESIGNS • COPYRIGHT • TRADE MARKS
The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

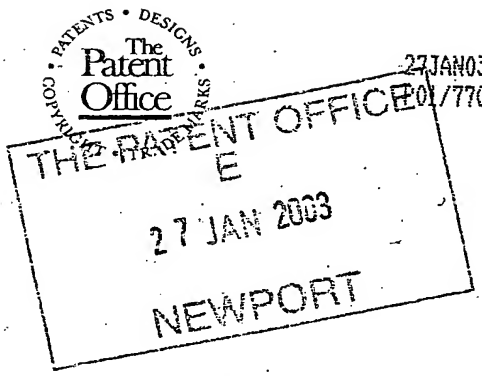
Signed D Evans

Dated 17 December 2003

THIS PAGE BLANK (USPTO)

Patents Form 1/77

Pat[illegible] Act 1977
(Rul[illegible]



27JAN03 E780035-1 D01463
01/7700 0.00-0301777.9

1/77

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

1. Your reference — 200207542-1 GB

2. Patent application number *(The Patent Office will fill in this part)* — 0301777.9

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Hewlett-Packard Company
3000 Hanover Street
Palo Alto
CA 94304, USA

Patents ADP number *(if you know it)*

If the applicant is a corporate body, give the country/state of its incorporation — Delaware, USA

7812985001

4. Title of the invention — Improvements in and Relating to Computer Operating System Data Management

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Richard A. Lawrence
Hewlett-Packard Ltd, IP Section
Filton Road, Stoke Gifford
Bristol BS34 8QZ

Patents ADP number *(if you know it)*

7448035001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*

a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	24
Claim(s)	9
Abstract	1
Drawing(s)	5+5

10. If you are also filing any of the following, state how many against each item.

Priority documents	-
Translations of priority documents	-
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	1
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	-
Any other documents *(please specify)*	Fee Sheet

11. I/We request the grant of a patent on the basis of this application.

Signature Richard A. Lawrence Date 27/1/03

12. Name and daytime telephone number of person to contact in the United Kingdom: Meg Joyce Tel: 0117-312-9068

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

Improvements in and Relating to Computer Operating System Data Management

The present invention relates to methods of computer operating system data management, to computing platforms for computer operating system data management, to computer programs including instructions configured to enable computer operating system data management, to computer operating systems arranged to perform operating system data management, to a computer operating system data management method, and, to computer operating system data management apparatus.

Data management is increasingly important as widespread access to public computer networks facilitates distribution of data. Distribution of data over public computer networks may be undesirable when the data in question comprises sensitive, confidential, copyright or other similar information.

A computer operating system can typically monitor input of data to a process or output of data by a process and apply appropriate management restrictions to these operations. Exemplary restrictions may prevent write operations to a public network, or to external memory devices for data having certain identifiable characteristics. However, manipulation of data within a process can not be monitored by the operating system. Such manipulation may modify the identifiable characteristics of data, and thus prevent the operating system from carrying out effective data management.

Particular problems arise when different types of data are assigned different levels of restriction, and processes involving data from different levels of restriction are run alongside one another. An operating system cannot guarantee that the different types of data have not been mixed. To maintain a desired level of restriction for the most restricted data in these circumstances, this level of restriction must be applied to all data involved in the processes. Consequently, data can only be upgraded to more restricted levels, leading to a system in which only highly trusted users/systems are allowed access to any data.

It is an aim of preferred embodiments of the present invention to overcome at least some of the problems associated with the prior art, whether identified herein, or otherwise.

According to a first aspect of the present invention there is provided a method of computer operating system data management, the method comprising the steps of: (a) associating data management information with data input to a process; and (b) regulating operating system operations involving the data according to the data management information.

By associating data management information at the operating system level greater security and flexibility is obtained; features that are often mutually exclusive.

Suitably, supervisor code administers the method by controlling the process at run time.

Suitably, the step (a) comprises associating data management information with data as the data is read into a memory space. Suitably, the step (a) comprises associating data management information with at least one data sub-unit as data is read into a memory space from a data unit comprising a plurality of data sub-units. Suitably, the step (a) comprises associating data management information with each independently addressable data unit that is read into the memory space. Suitably, the data management information is written to a data management memory space under control of the supervisor code. Suitably, the supervisor code comprises state machine automatons arranged to control the writing of data management information to the data management memory space.

Suitably, the step (b) comprises sub-steps (b1) identifying an operation involving the data; (b2) if the operation involves the data and is carried out within the process, maintaining an association between an output of the operation and the data management information; and (b3) if the operation involving the data includes a write operation to a location external to the process, selectively performing the operation dependent on the data management information.

Suitably, the step (b1) comprises: analysing process instructions to identify operations involving the data; and, providing instructions relating to the data management information with the operations involving the data. Suitably, the process instructions are analysed as blocks, each block defined by operations up to a terminating condition.

According to a second aspect of the present invention there is provided a computing platform for computer operating system data management, the computing platform comprising a data management unit, the data management unit arranged to associate data management information with data input to a process, and regulate operating system operations involving the data according to the data management information.

Suitably, the computing platform further comprises a memory space, and is arranged to load the process into the memory space and run the process under the control of the data management unit.

Suitably, the data management information is associated with at least one data sub-unit as data is input to a process from a data unit comprising a plurality of sub-units.

Suitably, data management information is associated with each independently addressable data unit.

Suitably, the data management unit comprises part of an operating system kernel space. Suitably the operating system kernel space comprises a tagging driver arranged to control loading of a supervisor code into the memory space with the process.

Suitably the supervisor code controls the process at run time to administer the operating system data management unit. Suitably, the supervisor code is arranged to analyse instructions of the process to identify operations

involving the data, and, provide instructions relating to the data management information with the operations involving the data.

Suitably, the memory space further comprises a data management information area under control of the supervisor code arranged to store the data management information.

Suitably, the data management unit comprises a data filter to identify data management information associated with data that is to be read into the memory space. The data filter may associate data management information with data read into the memory space from predetermined sources. The data filter may associate default data management information with data read into the memory space. Suitably, the data management unit further comprises a tag management module arranged to allow a user to specify data management information to be associated with data.

Suitably, the data management unit comprises a tag propagation module arranged to maintain an association with the data that has been read into the process and the data management information associated therewith. Suitably, the tag propagation module is arranged to maintain an association between an output of operations carried out within the process and the data management information associated with the data involved in the operations.

Suitably, the tag propagation module comprises state machine automatons arranged to maintain an association between an output of operations carried out within the

process and the data management information associated with the data involved in the operations.

According to a third aspect of the present invention there is provided a computer operating system data management method comprising the step of: identifying data having data management information associated therewith when the data is to be read into a memory space.

Suitably, the method further comprises the step of associating data management information with the data if the data is identified as having no data management information associated therewith.

Suitably, the data management information associated with data is read into the memory space with the data.

Suitably, the method further comprises the step of maintaining an association between the data and the data management information when the data is involved in operations within the process, and associating data management information with other data resulting from operations involving the data.

Suitably, the step of maintaining an association between the data and the data management information when the data is involved in operations within the process, and associating data management information with other data resulting from operations involving the data is carried out according to state machine automatons.

Suitably, the method further comprises the step of examining the data management information when the data is

to be involved in an operation external to the process, and allowing the operation if it is compatible with the data management information. Suitably, the operation is blocked if it is not compatible with the data management information.

Suitably, an operation external to the process may be compatible with the data management information subject to including the associated data management information with an output of the operation.

Suitably, the data management information identifies a set of permitted operations.

According to a fourth aspect of the present invention there is provided a computer operating system data management apparatus arranged to identify data having data management information associated therewith when data is read into a memory space.

Suitably, the data filter comprises part of a data management unit, and is arranged to associate data management information with the data if the data is identified as having no data management information associated therewith.

Suitably, the data management unit is arranged read the data management information associated with data is into the memory space with the data.

Suitably, the data management unit comprises a tag propagation module arranged to maintain an association between the data and the data management information when

the data is involved in operations within the process, and to associate data management information with other data resulting from operations involving the data.

Suitably, the tag propagation module comprises state machine automatons arranged to maintain an association between the data and the data management information when the data is involved in operations within the process, and to associate data management information with other data resulting from operations involving the data.

Suitably, the tag propagation module is arranged to examine the data management information when the data is to be involved in an operation external to the process, and cause the operation to be allowed if it is compatible with the data management information.

Suitably, the tag propagation module is arranged to cause the operation to be blocked if the operation is not compatible with the data management information.

Suitably, the tag propagation module is arranged to perform the operation external to the process subject to including the associated data management information with an output of the operation.

Suitably, the data management information identifies a set of permitted operations.

According to a fifth aspect of the present invention there is provided a computer program including instructions configured to enable computer operating system data

management in accordance with the first aspect of the invention.

According to a sixth aspect of the invention there is provided an operating system comprising an application code modifying unit arranged to perform a method of computer operating system data management in accordance with the first aspect of the invention.

For a better understanding of the invention, and to show how embodiments of the same may be carried into effect, reference will now be made, by way of example, to the accompanying diagrammatic drawings in which:

Figure 1 shows a computing platform for computer operating system data management according to a first embodiment of the invention;

Figure 2 shows a first operating system data management architecture suitable for use in the computing platform of Figure 1;

Figure 3 shows a second operating system data management architecture suitable for use in the computing platform of Figure 1; and

Figure 4 shows a flow diagram comprising steps involved in embodiments of the invention; and

Figure 5 shows a flow diagram comprising further steps involved in embodiments of the invention.

Data management in the form of data flow control can offer a high degree of security for identifiable data. Permitted operations for identifiable data form a security policy for that data. However, security of data management systems based on data flow control is compromised if applications involved in data processing can not be trusted to enforce the security policies for all data units and sub-units to which the applications have access. In this document, the term "process" relates to a computing process. Typically, a computing process comprises the sequence of states run through by software as that software is executed.

Figure 1 shows a computing platform 1 for computer operating system data management comprising, a processor 5, a memory space 10, an OS kernel space 20 comprising a data management unit 21 and a disk 30. The memory space 10 comprises an area of memory that can be addressed by user applications. The processor 5 is coupled to the memory space 10 and the OS kernel space 20 by a bus 6. In use, the computing platform 1 loads a process to be run on the processor 5 from the disk 30 into the memory space 10. It will be appreciated that the process to be run on the processor 5 could be loaded from other locations. The process is run on the processor under the control of the data management unit 21 such that operations involving data read into the memory space 10 by the process are regulated by the data management unit 21. The data management unit 21 regulates operations involving the data according to data management information associated with the data as it is read into the memory space 10.

The data management unit 21 propagates the data management information around the memory space 10 as process operations involving that data are carried out, and prevents the data management information from being read or written over by other operations. The data management unit includes a set of allowable operations for data having particular types of data management information therewith. By inspecting the data management information associated with a particular piece of data, the data management unit 21 can establish whether a desired operation is allowed for that data, and regulate the process operations accordingly.

Figure 2 shows an example operating system data management architecture comprising an OS kernel space and a memory space suitable for use in the computing platform of Figure 1. The example architecture of Figure 2 enables regulation of operations involving data read into a memory space by enforcing data flow control on applications using that data. The example architecture of Figure 2 relates to the Windows NT operating system. Windows NT is a registered trade mark of Microsoft Corporation.

Figure 2 shows a memory space comprising a user space 100 and an OS kernel space 200. The user space 100 comprises application memory spaces 110A,110B, supervisor code 120A,120B, and a tag table 130. The OS kernel space 200 comprises a standard NT kernel 250, file system driver 202 and storage device drivers 203. The OS kernel space 200 further comprises a tagging driver 210, a tag propagation module 220, and a tag management module 230 and a data filter 240.

When an application is to be run in the user space 100, information comprising the application code along with any required function libraries, application data etc. is loaded into a block of user memory space comprising the application memory space 110 under the control of the NT kernel 250. The tagging driver 210 further appends supervisor code to the application memory space 110 and sets aside a memory area for data management information. This memory area comprises the tag table 130.

In preference to allowing the NT kernel 250 to run the application code, the tagging driver 210 receives a code execution notification from the NT kernel 210 and runs the supervisor code 120

When run, the supervisor code 120 scans the application code starting from a first instruction of the application code, and continues through the instructions of the application code until a terminating condition is reached. A terminating condition comprisesan instruction that causes a change in execution flow of the application instructions., Example terminating conditions include jumps to a subroutines, interrupts etc. A portion of the application code between terminating conditions comprises a block of code.

The block of code is disassembled, and data management instructions are provided for any instructions comprising data read/writes to the memory, disk, registers or other functional units such as logic units, or to other input/output (I/O) devices. The data management instructions may include the original instruction that prompted provision of the data management instructions,

along with additional instructions relating to data management. Once a block of the application code has been scanned and modified, the modified code can be executed. The scanning process is then repeated, starting with the first instruction of the next block.

At a first system call of the application code relating to a particular piece of data, typically a read instruction, the first data management instruction associates data management information with the data. The data management information comprises a tag held in the tag table 130. The tag table 130 comprises a data management information memory area which can only be accessed by the supervisor code 120. Preferably, a tag is applied to each independently addressable unit of data - normally each byte of data. By applying a tag to each independently addressable piece of data all useable data is tagged, and, maximum flexibility regarding the association of data with a tag is maintained. A tag may preferably comprise a byte or other data unit.

A tag identifies a data management policy to be applied to the data associated with that tag. Different data management policies may specify a number of rules to be enforced in relation to data under that data management policy, for example, "data under this policy may not be written to a public network", or "data under this policy may only be operated on in a trusted environment". When independently addressable data units have their own tags it becomes possible for larger data structures such as e.g. files to comprise a number of independently addressable data units having a number of different tags. This ensures the correct policy can be associated with a particular data unit irrespective of its location or

association with other data in a memory structure, file structure or other data structure. The data management policy to be applied to data, and hence the tag, can be established in a number of ways.

(1) Data may already have a predetermined data management policy applied to it, and hence be associated with a pre-existing tag. When the NT kernel 250 makes a system call involving a piece of data, the data filter 240 checks for a pre-existing tag associated with that data, and if a pre-existing tag is present notifies the tag propagation module 220 to include the tag in the tag table 130, and to maintain the association of the tag with the data. Any tag associated with the data is maintained, and the data keeps its existing data management policy.

If there is no tag associated with the data, the following tag association methods can be used.

(2) Data read from a specific data source can have a predetermined data management policy corresponding to that data source applied to it. The data filter 240 checks for a data management policy corresponding to the specific data source, and if a predetermined policy does apply to data from that source notifies the tag propagation module 220 to include the corresponding tag in the tag table 130 and associate the tag with the data. For example, all data received over a private network from a trusted party can be associated with a tag indicative of the security status of the trusted party.

(3) When data has no pre-existing tag, and no predetermined data management policy applies to the data

source from which the data originates, the tag management module 230 initiates an operating system function that allows a user to directly specify a desired data management policy for the data. The desired data management policy specified by the user determines the tag associated with the data. To ensure that the operating system function is authentic and not subject to subversion, it is desired that the operating system function of the tag management module 230 is trusted. This trust can be achieved and demonstrated to a user in a number of ways, as will be appreciated by the skilled person.

(4) Alternatively, when data has no pre-existing tag, and no predetermined data management policy applies to the data source from which the data originates a default tag can be applied to the data.

Data management instructions are provided for subsequent instructions relating to internal processing of the tagged data. The data management instructions cause the tag propagation module 220 to maintain the association between the data and tag applied to it. Again, the data management instructions may include the instructions relating to internal processing of the data along with additional data management instructions. If the data is modified, e.g. by a logical or other operations, the relevant tag is associated with the modified data. Data management instructions for maintaining the association of tags with data as that data is manipulated and moved can be implemented using relatively simple state machine automatons. These automatons operate at the machine code level to effectively enforce the association and

propagation of tags according to simple rules. For example, if data is moved the tag associated with the data at the move destination should be the same as the tag associated with the data before the move. In this simple example, any tag associated with the data at the move destination can be overwritten by the tag associated with the incoming data. Other automatons can be used to combine tags, swap tags, extend tags to other data, leave tags unchanged etc. dependent on the existing data tag(s) and type of operation to be carried out on the data.

The supervisor code 120 manages the tags in the tag table. A simple form of tag management comprises providing a data tag table that is large enough to accommodate a tag for each piece of tagged data. This results in a one-to-one relationship between the data in the application memory space 110, and the data tags in the tag table, and a consequent doubling of the overall memory space required to run the application. However, memory is relatively cheap, and the one to one relationship enables simple functions to be used to associate the data with the relevant tag. As an alternative, different data structures can be envisaged for the data management information area, for example, a tag table can identify groups of data having a particular tag type. This may be advantageous when a file of data all associated with a single tag is involved in an operation. When more than one application is loaded in the user space 100, as shown in Figure 2 with the two application memory spaces 110A,110B, a shared tag table 130 can be used. As already mentioned, different tags can be applied to a separate data units within a file or other data structure. This allows an improved flexibility in subsequent manipulation

of the data structure ensuring the appropriate policy is applied to the separate data units.

Data management instructions are also provided for instructions relating to writing of data outside the process. The data management instructions may include the instructions relating to writing of data outside the process along with other data management instructions. In this case, the data management instructions prompt the supervisor code 120 to notify the tag propagation module 220 of the tag associated with the data to be written. The system call to the NT kernel 250 is received by the data filter 240. The data filter 240 queries the allowability of the requested operation with the tag propagation module 220 to verify the tag associated with the data to be written, and check that the data management policy identified by the tag allows the desired write to be performed with the data in question. If the desired write is within the security policy of the data in question, it is performed, with the data filter 240 controlling the file system driver 202 to ensure that the storage device drivers 203 to enforce the persistence of the tags with the stored data. If the data is not permitted to be written as requested, the write operation is blocked. Blocking may comprise writing random bits to the requested location, writing a string of zeros or ones to the requested location, leaving the requested location unaltered, or encrypting the data before writing.

A second example operating system data management architecture suitable for use in the computing platform of Figure 1 is shown in Figure 3. The example operating

system data management architecture of Figure 3 relates to the Linux operating system.

Figure 3 shows a user space 100 and an OS kernel space 200. The user space 100 comprises application memory spaces 110A,110B, supervisor code 120A,120B, and a tag table 130. The OS kernel space 200 comprises a tag propagation module 220, a tag management module 230, along with a Linux kernel 260 comprising an executable loader module 261, a process management module 262, a network support module 263 and a file system support module 264.

As the Linux operating system is open source, a number of the functions required to implement the data management system can be incorporated into the existing functional blocks of the kernel. In the example architectures of Figure 3, the executable loader module 261, the process management module 262, the network support module 263 and the file system support module 264 are be modified versions of those included in a standard Linux kernel, as will be described below.

As before, the supervisor code 120 controls system calls, handles memory space tag propagation, and instructs policy checks in the OS kernel space 200 when required. Also as before, the tag propagation module 220 maintains policy information relating to allowable operations within the policies, and the tag management module 230 provides an administrative interface comprising an operating system function that allows a user to directly specify a desired data management policy for the data.

The operation of the Linux kernel 260 allows the data management architectures shown to carry out data flow control. The executable loader 261 includes a tagging driver that ensures applications are run under the control of the supervisor code 120. The process management module 262 carries out process management control to maintain the processor running the application or applications in a suitable state to enable tag association, monitoring and propagation. The network support module 263 enables the propagation of tags with data across a network, and the file system support module 264 enables the propagation of tags with data on disk. The network support module 263 and the file system support module 264 together provide the functionality of the data filter of Figure 2. Again, state machine based automation can be used to perform basic tag association, monitoring and propagation functions at a machine code level.

The modifications to the executable loader module 261, the process management module 262, the network support module 263 and the file system support module 264 can be easily implemented with suitable hooks.

Figure 4 shows a flow diagram outlining basic steps in an example method of operating system data management.

The method comprises a first step 300 of associating data management information with data input to a process; and a second step 310 of regulating operations involving the data input to the process in the first step 300 according to the data management information associated with the data in the first step 300. The basic first and second

steps 300,310 are further expanded upon in the flow diagram of Figure 5.

Figure 5 shows a flow diagram outlining further steps in an example method of operating system data management.

The method of Figure 5 starts with an "external operation?" decision 312. If data on which the method is performed is read into memory space associated with a process from a location external to the memory space associated with the process, the outcome of the "external operation?" decision 312 is YES. Furthermore, if the data within the process is to be written to an external location, the outcome of the "external operation?" decision 312 is also YES. Following a positive decision at the "external operation?" decision, the method moves to the "tag present?" decision 314. Operations involving data within the process result in a negative outcome at the "external operation?" decision 312.

At the "tag present?" decision 314, it is determined whether the data involved in the operation has data management information associated with it. If the data has no data management information associated with it, the association step 300 is performed, and the method returns to the "external operation?" decision 312.

In the association step 300, data management information is associated with the data in question. This association can be carried out by any of the methods described earlier, or by other suitable methods.

Following a positive decision at the "tag present?" decision 314, the method moves to the "operation allowed?" decision 316. At this decision, the data management information associated with the data is examined, and its compatibility with the specified external operation identified in the "external operation?" decision 312 is established.

If the data management information is compatible with the external operation, it is carried out in the execution step 318. Following the execution step 318, the method returns to the "external operation?" decision 312. Alternatively, if the data management information is not compatible with the external operation, it is blocked in the blocking step 318. Blocking in step 318 can comprise any of the methods described earlier, or by other suitable methods.

Any operations identified at the "external operation?" decision 312 as internal operations are carried out, with association of the data involved in the operation with the relevant data management information maintained in the tag propagation step 313.

Including the data management functionality with an operating system provides a first level of security, as operating system operation should be relatively free from security threatening bugs compared to either commercial or open source application software. Furthermore, if the operating system allows trusted operation after a secure boots, for example as provided for by the Trusted Computing Platform Alliance (TCPA) standard, the data management functionality can also form part of the trusted

system. This enables the data management functions to also form part of the trusted system, enabling e.g. digital rights management or other secrecy conditions to be enforced on data.

It is possible that the computing platform for operating system data management could refuse to open or write data with a pre-existing tag unless the computing platform is running in a trusted mode, adding to the enforceability of data flow control under the data management system. This is particularly useful when encrypted data is moved between trusted computing platforms over a public network.

An operating system running as a virtual machine using an aspect of the present invention, also falls within its scope.

An operating system data management method and a computing platform for operating system data management have been described. The data management method and computing platform allow a supervisor code to monitor data flow into and out of an application using data management information. As data is used within an application process, the data management information is propagated with the data. This allows the supervisor code to ensure that only external write operations which are compatible with a data management policy for the data are performed. The data flow monitoring and enforcement enabled by the data management method and computing platform facilitate the construction of systems that support digital rights management and other data privacy functions, but avoid the problems associated with system wide approaches to data flow control systems. In particular, the granularity

provided by associating data management information with data units that are individually addressable rather than with a data structure such as a file of which the individually addressable data units are part offers improved flexibility in how security is enforced. The method and computing platform described do not require source code modification of application and subsequent recompilation. Furthermore, the method and system described can easily be retrospectively implemented in a variety of known operating systems, for example Windows NT and Linux as show herein.

The reader's attention is directed to all papers and documents which are filed concurrently with or previous to this specification in connection with this application and which are open to public inspection with this specification, and the contents of all such papers and documents are incorporated herein by reference.

All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or steps are mutually exclusive.

Each feature disclosed in this specification (including any accompanying claims, abstract and drawings), may be replaced by alternative features serving the same, equivalent or similar purpose, unless expressly stated otherwise. Thus, unless expressly stated otherwise, each feature disclosed is one example only of a generic series of equivalent or similar features.

The invention is not restricted to the details of the foregoing embodiment(s). The invention extends to any novel one, or any novel combination, of the features disclosed in this specification (including any accompanying claims, abstract and drawings), or to any novel one, or any novel combination, of the steps of any method or process so disclosed.

Claims

1. A method of computer operating system data management comprising the steps of:

(a) associating data management information with data input to a process; and

(b) regulating operating system operations involving the data according to the data management information.

2. The method of claim 1 wherein supervisor code administers the method by controlling the process at run time.

3. The method of claim 1 or 2, wherein, the step (a) comprises associating data management information with data as the data is read into a memory space.

4. The method of any preceding claim wherein the step (a) comprises associating data management information with at least one data sub-unit as data is read into a memory space from a data unit comprising a plurality of data sub-units.

5. The method of any preceding claim wherein, the step (a) comprises associating data management information with each independently addressable data unit that is read into the memory space.

6. The method of claim 2 or any one of claims 3-5 as dependent on claim 2, wherein the data management

information is written to a data management memory space under control of the supervisor code.

7. The method of claim 6 wherein the supervisor code comprises state machine automatons arranged to control the writing of data management info to the data management memory space.

8. The method of any preceding claim wherein the step (b) comprises sub-steps (b1) identifying an operation involving the data; (b2) if the operation involves the data and is carried out within the process, maintaining an association between an output of the operation and the data management information; and (b3) if the operation involving the data includes a write operation to a location external to the process, selectively performing the operation dependent on the data management information.

9. The method of claim 8, wherein the step (b1) comprises: analysing process instructions to identify operations involving the data; and, providing instructions relating to the data management information with the operations involving the data.

10. The method of claim 9, wherein the process instructions are analysed as blocks, each block defined by operations up to a terminating condition.

11. A computing platform for operating system data management, the computing platform comprising a data management unit, the data management unit arranged to associate data management information with data input to a

process, and regulate operating system operations involving the data according to the data management information.

12. The computing platform of claim 11, further comprising a memory space, the computing platform arranged to load the process into the memory space and run the process under the control of the data management unit.

13. The computing platform of claim 11 or 12, wherein the data management information is associated with at least one data sub-unit as data is input to a process from a data unit comprising a plurality of sub-units.

14. The computing platform of any one of claims 11-13, wherein the data management information is associated with each independently addressable data unit.

15. The computing platform of any one of claims 11-14, wherein the data management unit comprises part of an operating system kernel space.

16. The computing platform of claim 15, wherein the operating system kernel space comprises a tagging driver arranged to control loading of a supervisor code into the memory space with the process.

17. The computing platform of claim 16, wherein the supervisor code controls the process at run time to administer the operating system data management unit.

18. The computing platform of claim 15 or 16, wherein the supervisor code is arranged to analyse instructions of the

process to identify operations involving the data, and, providing instructions relating to the data management information with the operations involving the data.

19. The computing platform of any of claims 16 to 18, wherein the memory space further comprises a data management information area under control of the supervisor code arranged to store the data management information.

20. The computing platform of any of claims 12 to 19, wherein the data management unit comprises a data filter arranged to identify data management information associated with data that is to be read into the memory space.

21. The computing platform of claim 20, wherein the data filter is arranged to associate data management information with data read into the memory space from predetermined sources, or alternatively is arranged to associate default data management information with data read into the memory space.

22. The computing platform of any of claims 11 to 21, wherein the data management unit further comprises a tag management module arranged to allow a user to specify data management information to be associated with data.

23. The computing platform of any of claims 11 to 22, wherein the data management unit comprises a tag propagation module arranged to maintain an association with the data that has been read into the process and the data management information associated therewith.

24. The computing platform of claim 23, wherein the tag propagation module is arranged to maintain an association between an output of operations carried out within the process and the data management information associated with the data involved in the operations.

25. The computing platform of claim 24, wherein the tag propagation module comprises state machine automatons arranged to maintain an association between an output of operations carried out within the process and the data management information associated with the data involved in the operations.

26. An operating system data management method comprising the step of: identifying data having data management information associated therewith when the data is to be read into a memory space.

27. The method of claim 26, further comprising the step of: associating data management information with the data if the data is identified as having no data management information associated therewith.

28. The method of claim 26 or 27, wherein the data management information associated with data is read into the memory space with the data.

29. The method of any of claims 26 to 28, further comprising the step of: maintaining an association between the data and the data management information when the data is involved in operations within the process, and

associating data management information with other data resulting from operations involving the data.

30. The method of claim 29, wherein the step of an association between the data and the data management information when the data is involved in operations within the process, and associating data management information with other data resulting from operations involving the data.

31. The method of claim 29 or 30, further comprising the step of: examining the data management information when the data is to be involved in an operation external to the process, and allowing the operation if it is compatible with the data management information.

32. The method of claim 31, wherein the operation is blocked if it is not compatible with the data management information.

33. The method of claim 31 or 32, wherein the operation external to the process is compatible with the data management information subject to including the associated data management information with an output of the operation.

34. The method of any of claims 26 to 33, wherein the data management information identifies a set of permitted operations.

35. An operating system data management apparatus comprising a data filter arranged to identify data having

data management information associated therewith when that data is read into a memory space.

36. The apparatus of claim 35, wherein the data filter comprises part of a data management unit, and is arranged to associate data management information with the data if the data is identified as having no data management information associated therewith.

37. The apparatus of claim 35 or 36, wherein data management unit is arranged read the data management information associated with data is into the memory space with the data.

38. The apparatus of any of claims 35 to 37, wherein the data management unit comprises a tag propagation module arranged to maintain an association between the data and the data management information when the data is involved in operations within the process, and to associate data management information with other data resulting from operations involving the data.

39. The apparatus of claim 38 wherein the tag propagation module comprises state machine automatons arranged to maintain an association between the data and the data management information when the data is involved in operations within the process, and to associate data management information with other data resulting from operations involving the data.

40. The apparatus of claim 38 or 39, wherein the tag propagation module is arranged to examine the data

management information when the data is to be involved in an operation external to the process, and cause the operation to be allowed if it is compatible with the data management information.

41. The apparatus of claim 40, wherein the tag propagation module is arranged to cause the operation to be blocked if the operation is not compatible with the data management information.

42. The apparatus of claim 40 or 41, wherein the tag propagation module is arranged to perform the operation external to the process subject to including the associated data management information with an output of the operation.

43. The apparatus of any of claims 35 to 42, wherein the data management information identifies a set of permitted operations.

44. A computer program including instructions configured to enable operating system data management in accordance with the method of operating system data management of claims 1 to 9, or the operating system data management method of claims 26 to 34.

45. An operating system comprising an application code modifying unit arranged to perform the method of operating system data management of any of claims 1 to 9 or the operating system data management method of claims 26 to 34.

46. A method of operating system data management substantially as herein described, with particular reference to Figures 4 and 5.

47. A computing platform for operating system data management substantially as herein described with particular reference to Figures 1 to 3.

Abstract

Improvements in and Relating to Computer Operating System Data Management

A method of computer operating system data management comprising the steps of: (a) associating data management information with data input to a process (300); and (b) regulating operating system operations involving the data according to the data management information is provided (310). A computing platform (1) for operating system data management is also provided. Furthermore, a computer program including instructions configured to enable operating system data management, an operating system, and an operating system data management method and apparatus arranged to identify data having data management information associated therewith when that data is read into a memory space are provided.

[Figure 1]

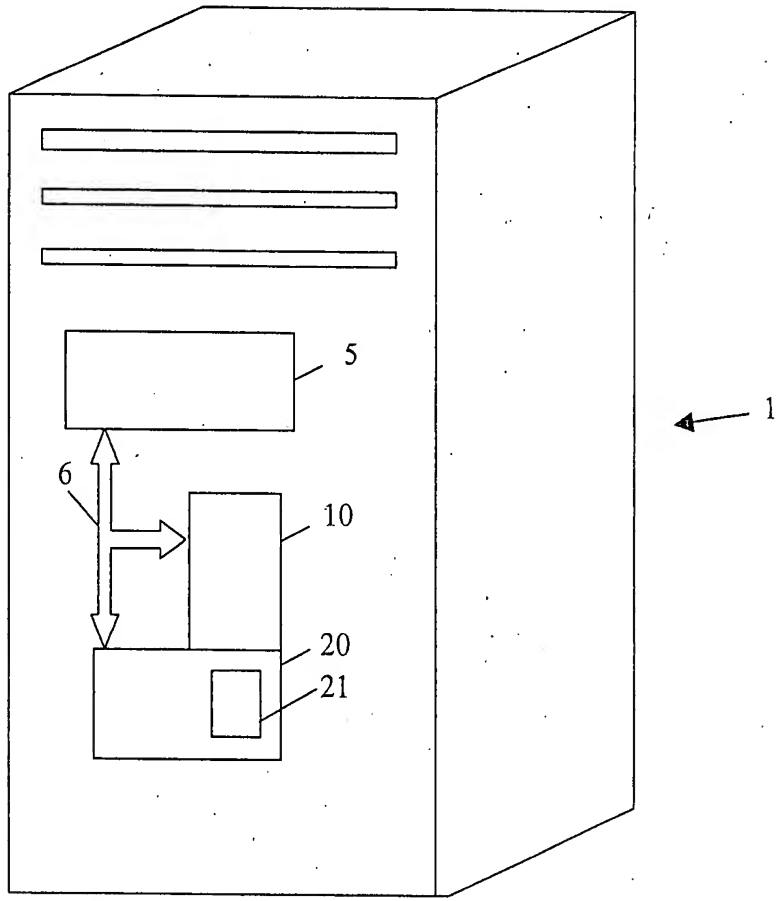


Figure 1

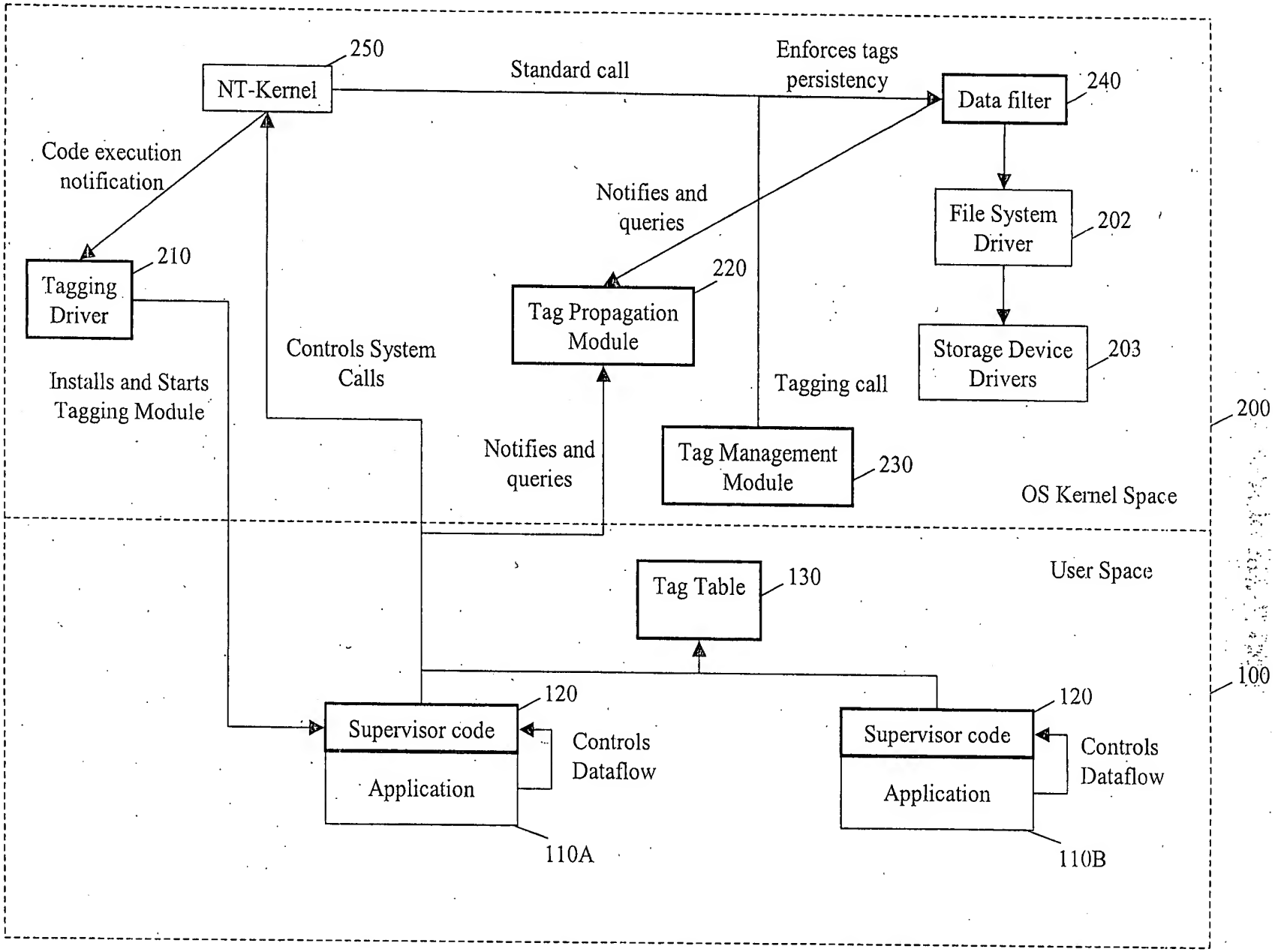

NT-Kernel
250
Standard call
Enforces tags persistency
Data filter
240
Code execution notification
Notifies and queries
File System Driver
202
Tagging Driver
210
220
Tag Propagation Module
Storage Device Drivers
203
Controls System Calls
Installs and Starts Tagging Module
Tagging call
Notifies and queries
Tag Management Module
230
200
OS Kernel Space
User Space
Tag Table
130
100
120
Supervisor code
Controls Dataflow
Application
110A
120
Supervisor code
Controls Dataflow
Application
110B

Figure 2

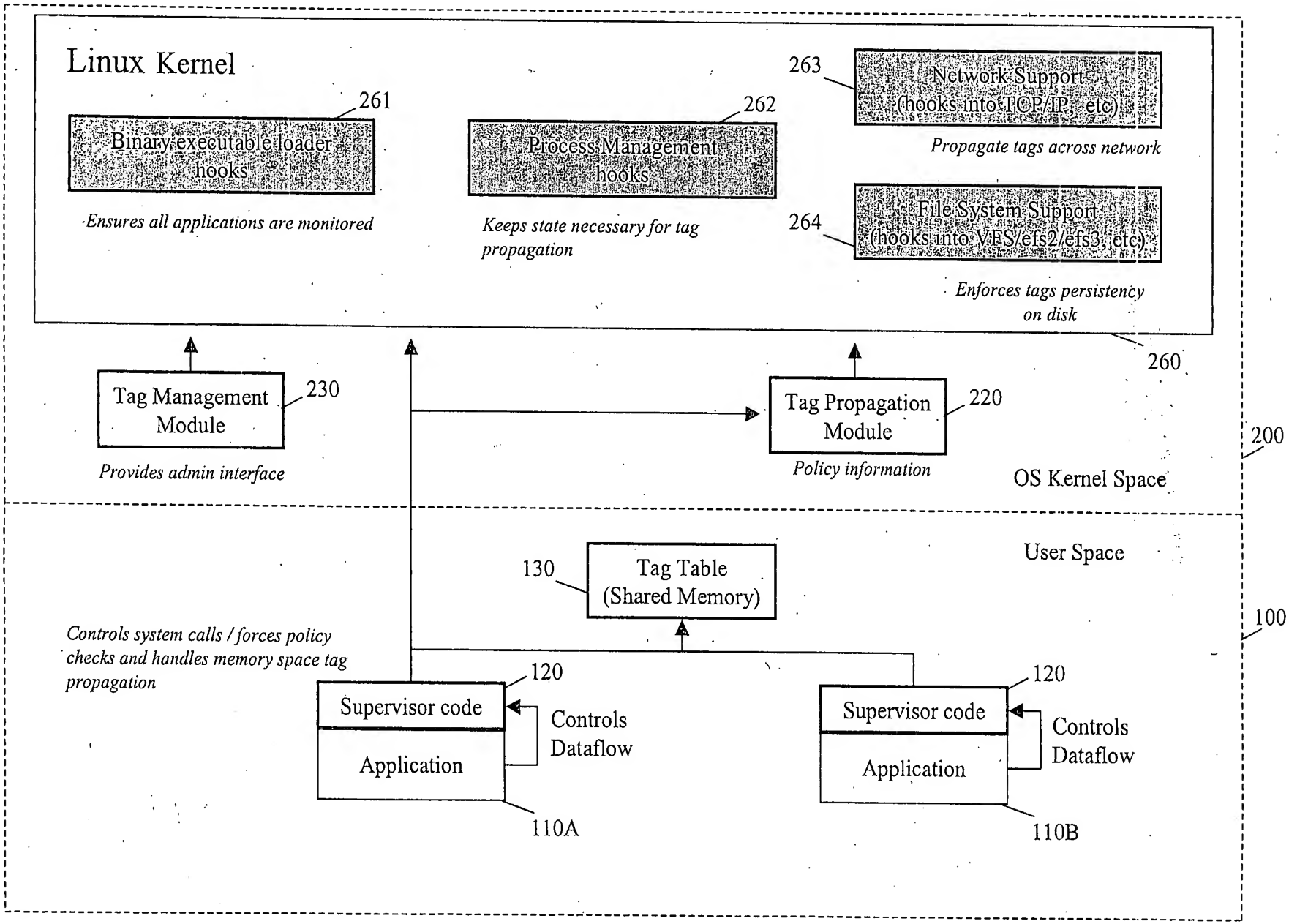

Linux Kernel
261
Binary executable loader hooks
Ensures all applications are monitored
262
Process Management hooks
Keeps state necessary for tag propagation
263
Network Support (hooks into TCP/IP, etc)
Propagate tags across network
264
File System Support (hooks into VFS/efs2/efs3, etc)
Enforces tags persistency on disk
260
Tag Management Module
230
Provides admin interface
Tag Propagation Module
220
Policy information
200
OS Kernel Space
User Space
130
Tag Table (Shared Memory)
100
Controls system calls / forces policy checks and handles memory space tag propagation
120
Supervisor code
Application
Controls Dataflow
110A
120
Supervisor code
Application
Controls Dataflow
110B

Figure 3

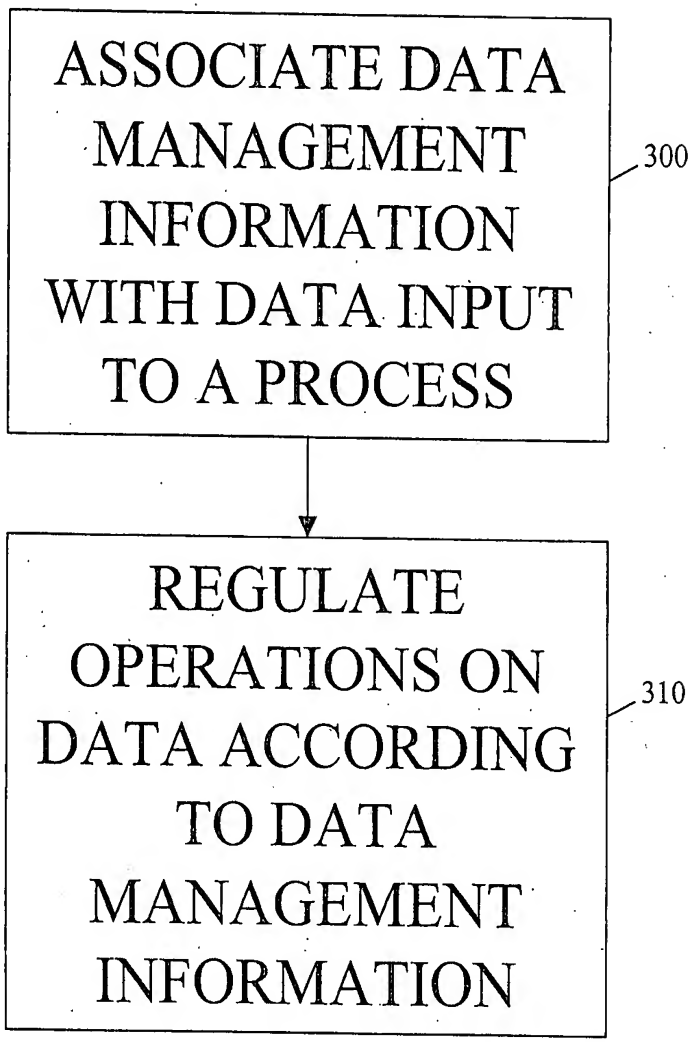

ASSOCIATE DATA MANAGEMENT INFORMATION WITH DATA INPUT TO A PROCESS
300
REGULATE OPERATIONS ON DATA ACCORDING TO DATA MANAGEMENT INFORMATION
310

Figure 4

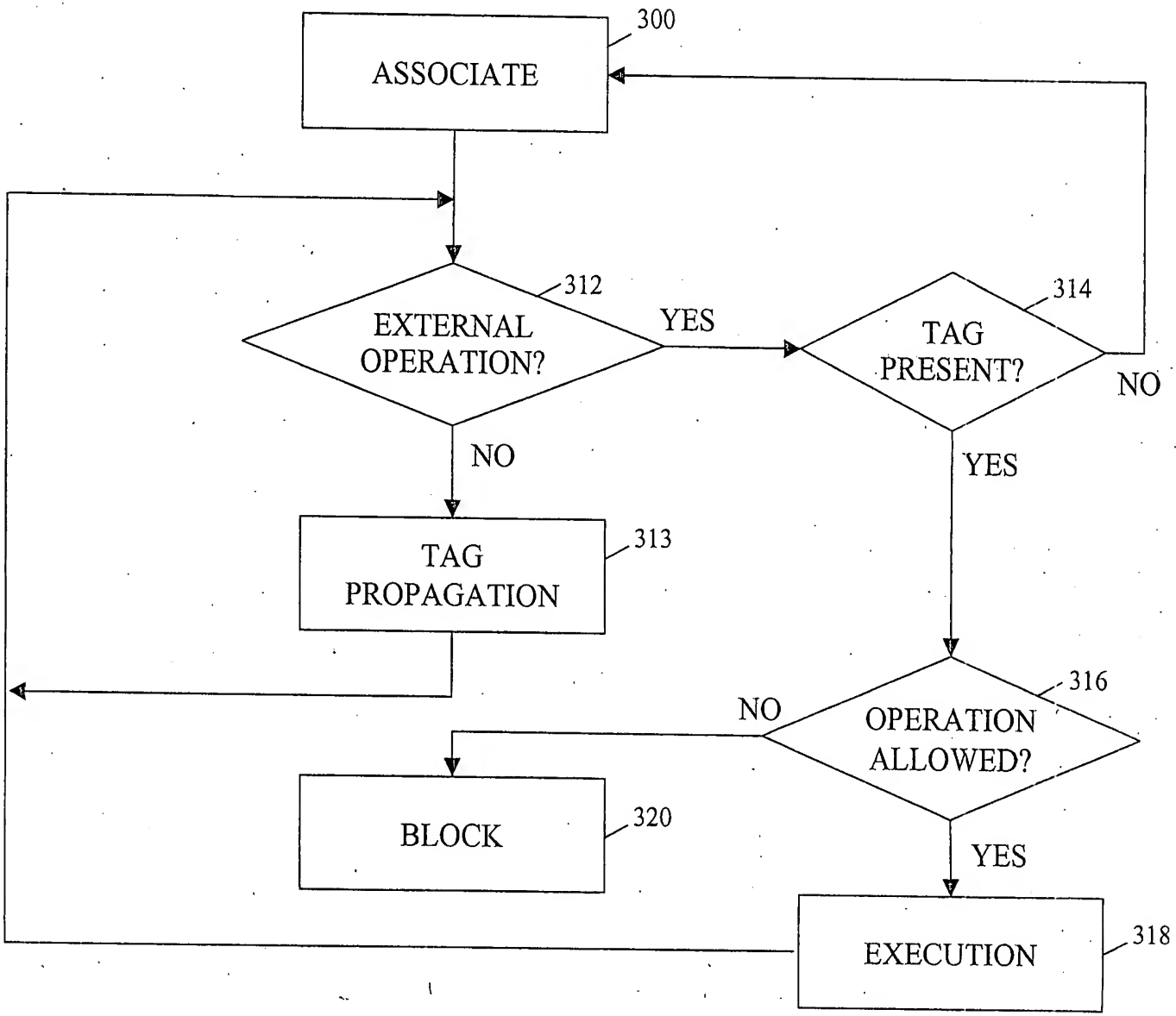

ASSOCIATE
300
EXTERNAL OPERATION?
312
YES
TAG PRESENT?
314
NO
NO
YES
TAG PROPAGATION
313
OPERATION ALLOWED?
316
NO
BLOCK
320
YES
EXECUTION
318

Figure 5

US 1076401702P1

Creation date: 02-02-2004
Indexing Officer: SGEBREHIWOT - SARA GEBREHIWOT
Team: OIPEScanning
Dossier: 10764017

Legal Date: 01-23-2004

No.	Doccode	Number of pages
1	NPL	66

Total number of pages: 66

Remarks:

Order of re-scan issued on .....